

## J2EE 1.3 – Introduction

### Part 1 - Packaging, EJB 2.0 Local Interfaces and CMP

WebSphere software



Updated 2/12/2003

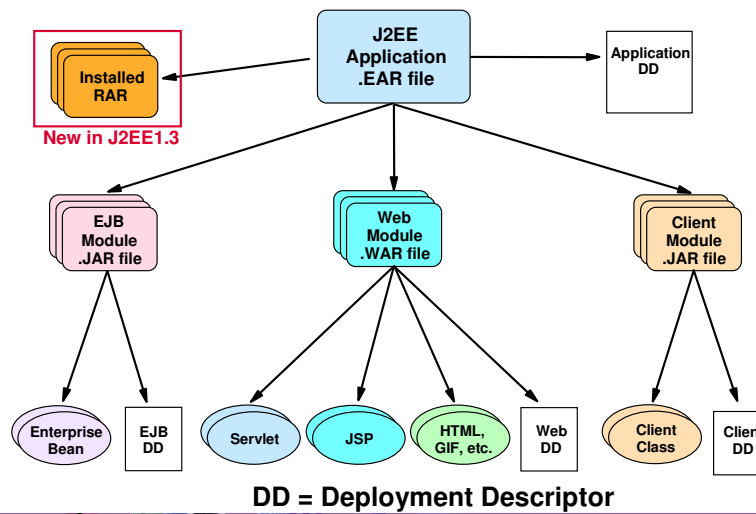
© 2002, 2003 IBM Corporation

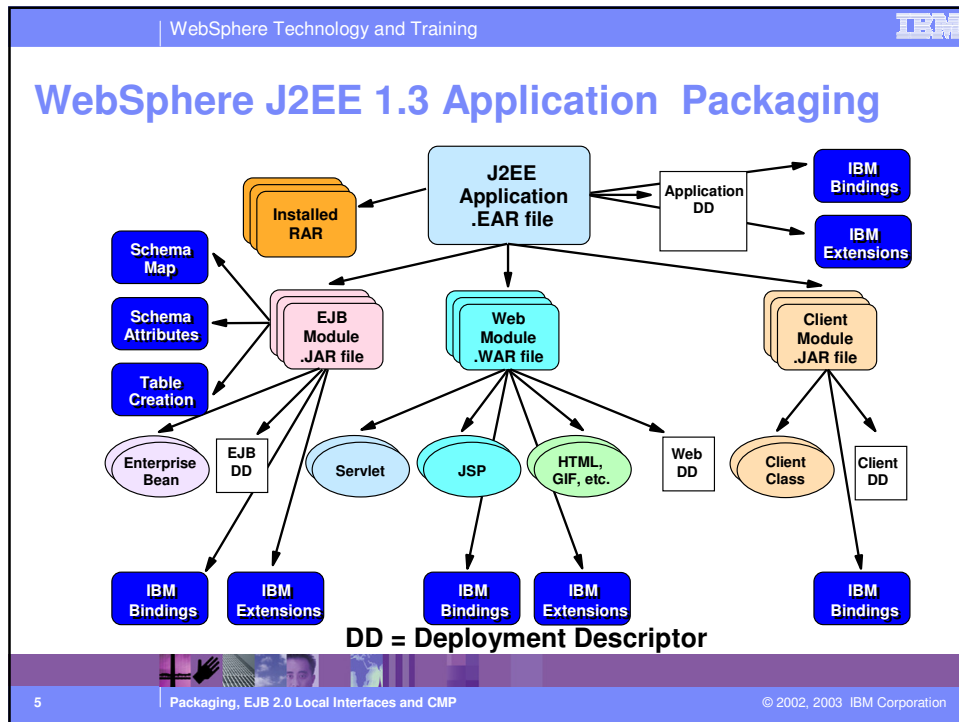
## J2EE 1.3 Topics


- ■ **J2EE 1.3 Packaging**
- **EJB 2.0:**
  - ■ **EJB 2.0 Interoperability**
  - ■ **New type of Interface: Local Interfaces**
  - ■ **New Persistence Manager to handle Container-Managed Persistence and Relationships**
    - EJB Query Language (EJB QL)
    - New type of bean: Message-driven Bean
    - EJB Home Methods
    - Dependent Values
- **Value-Add Features beyond EJB 2.0 specification**
- **J2EE 1.3 aspects of Web Components**
  - Servlets 2.3
  - HTTP Session Topics
  - JSP 1.2

# J2EE 1.3 Packaging and EJB Interoperability

## J2EE 1.3 Enterprise Application Packaging





WebSphere Technology and Training 

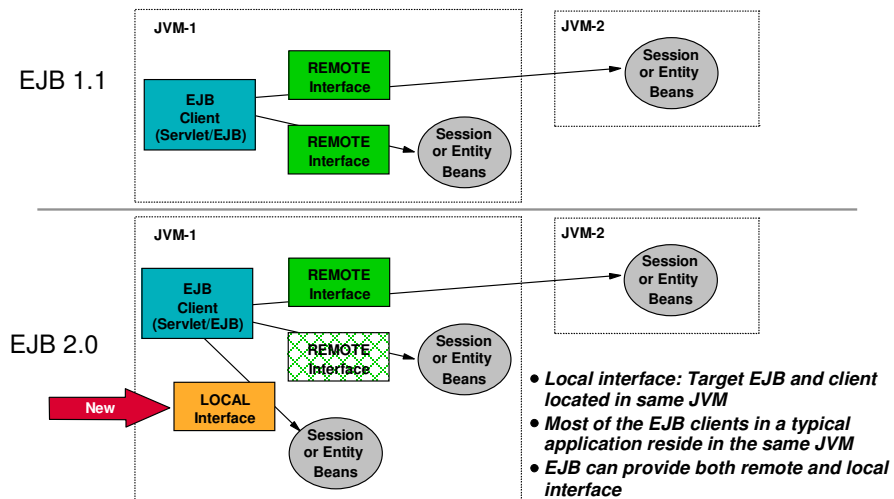
## EJB 2.0 Multi-vendor Interoperability

- **EJB 2.0 supports interoperability among EJB containers from different vendors**
- **Areas of interoperability addressed by the specs:**
  - Remote method invocation
    - Data type mapping
    - HandleDelegate - allows portable serialization/deserialization of EJBObject and EJBHome
  - Transaction Interop
  - Naming
    - CORBA CosNaming APIs support
  - Security
    - CSiv2

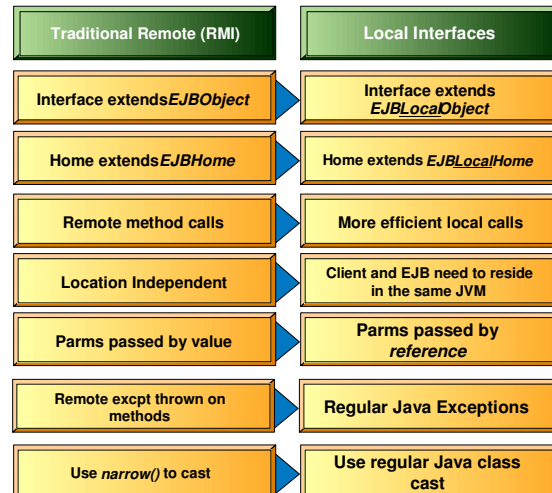
6 | Packaging, EJB 2.0 Local Interfaces and CMP © 2002, 2003 IBM Corporation

## EJB 2.0 Local Interface

### EJB 2.0: Interface comparison with EJB 1.1



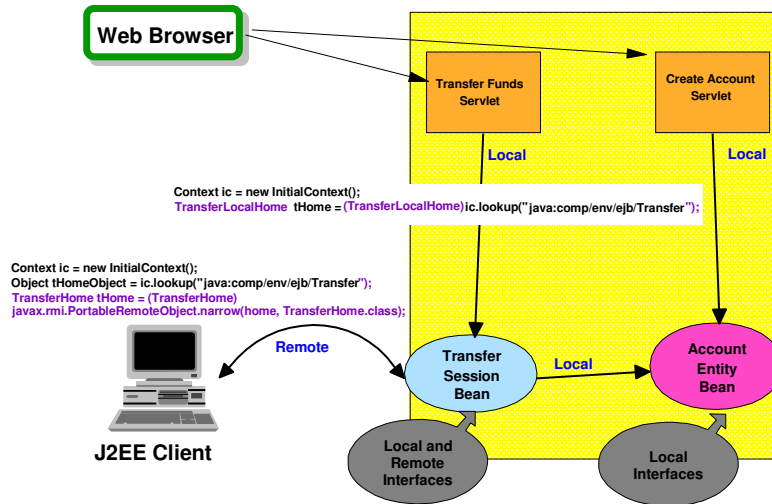
## Local and Remote Interface Comparison



## Local Interfaces vs. "No Local Copies"

- **WebSphere 3.5.x and 4.0 already "optimize" local EJB calls**
  - Through a simplified RMI/IIOP call
  - Through "No Local Copies" setting
- **Local Interfaces are conceptually different**
  - Local access is exposed at the programming model level
  - NOT an administrative option
  - Provide top efficiency for EJB calls, no RMI/IIOP involved
  - Local/remote transparency is not available

## EJB 2.0: Local Interface Example



## EJB 2.0 Local Interface - Deployment Descriptor

### Local Interface defined in EJB Deployment Descriptor (ejb-jar.xml)

```
<enterprise-beans>
  <entity id="Transfer">
    ...
    <home>com.ibm.examples.mybank.ejb.TransferHome</home>
    <remote>com.ibm.examples.mybank.ejb.Transfer</remote>
    ➔ <local-home>com.ibm.examples.mybank.ejb.TransferLocalHome</local-home>
    <local>com.ibm.examples.mybank.ejb.TransferLocal</local>
    ...
  </entity>
```

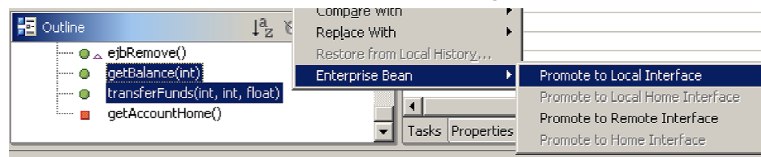
### ...and in the EJB client DD:

```
➔ <ejb-local-ref id="EjbRef_1">
  <description>Account Entity Bean</description>
  <ejb-ref-name>bank/Account</ejb-ref-name>
  <ejb-ref-type>Entity</ejb-ref-type>
  ➔ <local-home>com.ibm.examples.mybank.ejb.AccountLocalHome</local-home>
  ➔ <local>com.ibm.examples.mybank.ejb.AccountLocal</local>
  </ejb-local-ref>
```

## Tooling Support for Local/Remote Interfaces

**Local and/or Remote Interfaces specified when enterprise bean created with WebSphere Studio Application Developer**

**Promote methods to Local or Remote existing interfaces**

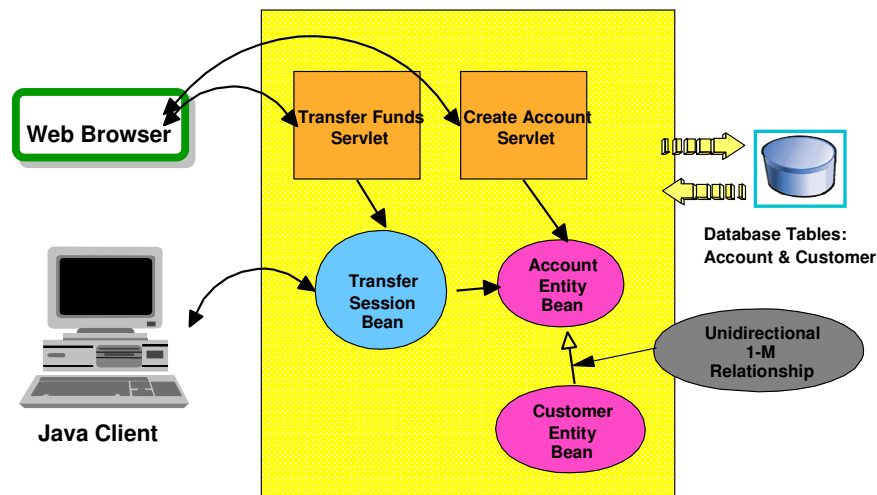


## EJB 2.0 Container-Managed Persistence Support

## EJB 2.0 - New Container Managed Persistence

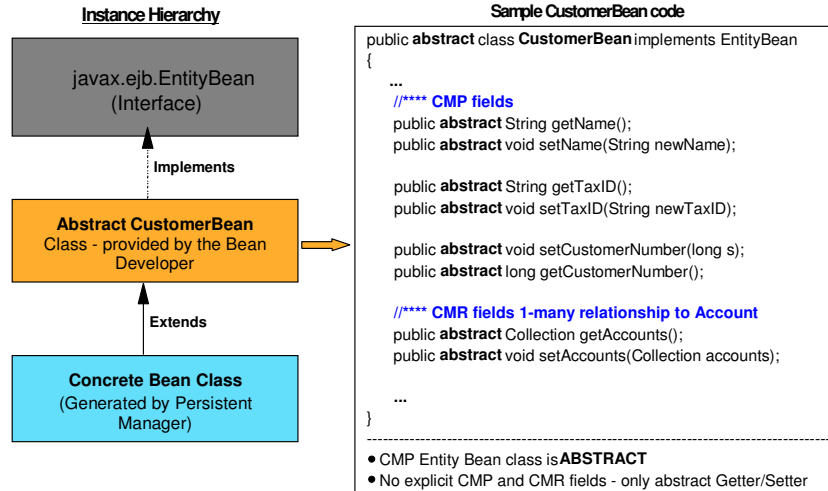
- **In EJB 1.1, persistent data were defined by Bean's instance variables**
  - Mostly not enough support persisting relationships with other beans
- **EJB 2.0 introduces abstract persistence schema**
  - Concrete implementation up to the container tools and runtime
  - Bean provider only responsible for defining abstract accessors to persistent data
- **EJB 2.0 CMP bean class**
  - Bean declared as abstract class
  - Persistent fields and relationships defined through abstract accessor methods (getter/setter)
  - Persistence Manager generates concrete implementation of the abstract bean class
    - Based on the XML deployment descriptor and the bean class
- **Container Managed Relationships (CMR)**
  - Allows multiple entity beans to have relationships among themselves
  - Container implements and supports the relationship
    - One-to-One, One-to-Many and Many-to-Many relationships
    - Uni- and bi-directional relationships
- **Advantages:**
  - More versatile container managed persistence approach
  - Opens the door to non-relational data stores

## EJB 2.0: CMP and CMR Example





## EJB 2.0: CMP Entity Bean Example



## EJB 2.0 CMP and CMR - Deployment Descriptor

- **CMP/CMR fields defined in EJB Deployment Descriptor (ejb-jar.xml)**

```

<enterprise-beans>
  <entity id="Customer">
    <ejb-name>Customer</ejb-name>
    ...
    <abstract-schema-name>Customer</abstract-schema-name>
    <cmp-field id="Customer_number">
      <field-name>customerNumber</field-name>
    </cmp-field>
    <cmp-field id="Customer_name">
      <field-name>name</field-name>
    </cmp-field>
    <cmp-field id="Customer_taxID">
      <field-name>taxID</field-name>
    </cmp-field>
    ...
  </entity>
  ...
</enterprise-beans>
  
```

CMP fields

## EJB 2.0: CMR - Example

```

<relationships id="Relationships_1">
  <ejb-relation id="EJBRelation_1">
    <ejb-relation-name>CustomerToAccounts</ejb-relation-name>
    <ejb-relationship-role id="EJBRelationshipRole_1">
      <ejb-relationship-role-name>OwnerOfAccounts</ejb-relationship-role-name>
      <multiplicity>One</multiplicity>
      <relationship-role-source id="RoleSource_1">
        <ejb-name> Customer </ejb-name>
      </relationship-role-source>
      <cmr-field id="CMRField_1">
        <cmr-field-name>accounts</cmr-field-name>
        <cmr-field-type>java.util.Collection</cmr-field-type>
      </cmr-field>
    </ejb-relationship-role>
    <ejb-relationship-role id="EJBRelationshipRole_2">
      <ejb-relationship-role-name>OwnedAccounts</ejb-relationship-role-name>
      <multiplicity>Many</multiplicity>
      <relationship-role-source id="RoleSource_2">
        <ejb-name> Account </ejb-name>
      </relationship-role-source>
    </ejb-relationship-role>
  </ejb-relation>
</relationships>

```

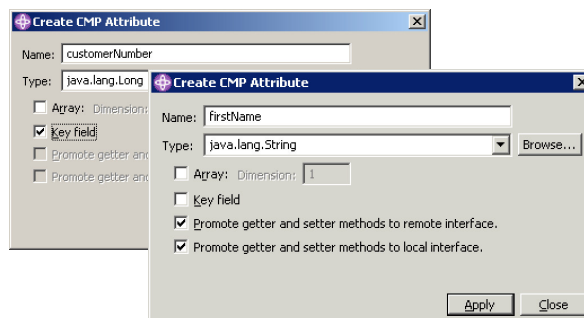
One side of the Relation - "Customer"

Other side of the Relation - "Account"

**One-Many Relationship between "Customer" and "Account"**  
**One Customer can have Many Accounts**

## CMP 2.0 Tool Support

- **CMP fields defined with EJB creation wizard in WebSphere Studio Application Developer**
- **Getters and Setters can be created for Non-Key fields when the enterprise bean is created**



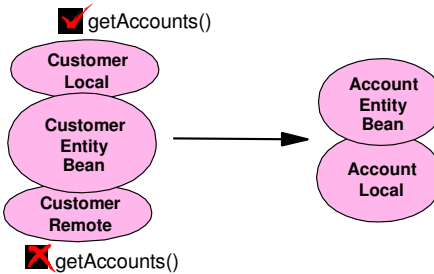
## CMR and Local Interfaces

### ■ "Target" EJB in a relationship must provide Local Interfaces

- In bi-directional relationships, both EJBs must have Local Interfaces

### ■ CMR accessor methods deal with Local Interfaces

- Must **not** be exposed on Remote Interfaces
- Remote clients can't get directly to related objects
  - Bean provider must add business logic to enable remote accessibility



## Tool Support for CMR

Relationships can only be created between Enterprise beans of the same specification (Account 2.0 EJB and Customer 2.0 EJB)

Multiplicity settings of One or Many

Multiplicity and Navigability allow access to Entity Beans in the relationship

Remove entity beans in relationship when delete called

## Summary

- **J2EE 1.3 specification supports adding Resource Adapter (RAR) files to EAR files**
- **Local Interfaces are a design decision which offers benefits in certain situations**
- **CMP in EJB 2.0 specification leaves implementation details to the Container and reduces requirement of tight integration with datastore**
- **CMR allows for interaction between entity beans corresponding to realistic business relationships**

This page is intentionally left blank.